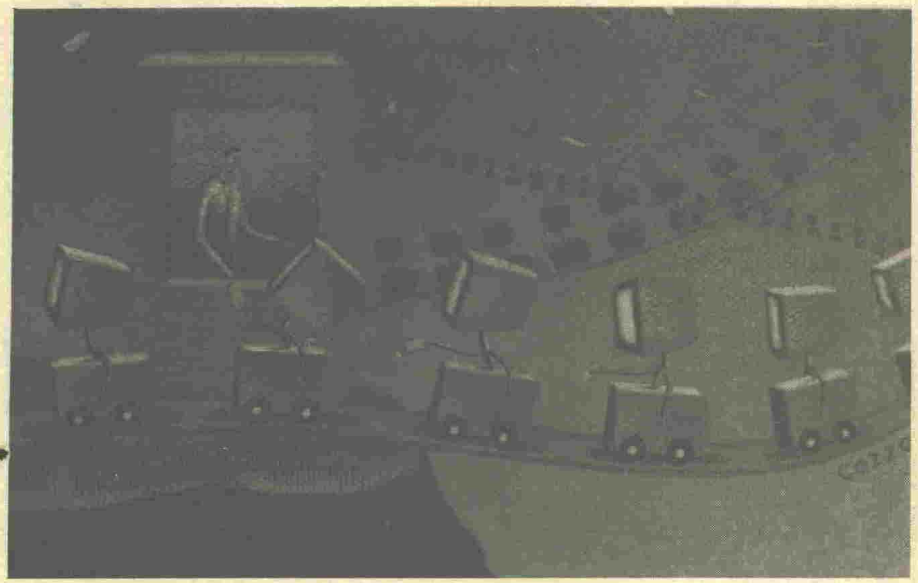
ยืน ภู่วรวรรณ

การกำหนดเส้นทาง ในระบบเครือข่ายคอมพิวเตอร์



อุปกรณ์ที่ทำหน้าที่นำข้อมูลดาต้าแกรมนำส่งที่สำคัญคืออุปกรณ์หาเส้นทางหรือเราเตอร์ การหาเส้นทางใช้โปรโตคอลสนับสนุนสองประการคือ RIP และ OSPF

เมื่อมีการสร้างเครือข่ายเชื่อมโยงกันมาก ๆ ก็จำเป็นจะต้องหาทางสร้างโปรโตคอลเพื่อการเชื่อมโยงหรือนำส่งข้อมูลให้ได้ดี จากที่ทราบกันแล้วว่า โปรโตคอล TCP/IP มีกลไกในการนำข้อมูลจากโปรแกรมประยุกต์ใส่ใน TCP แพคเก็ต จากนั้นใส่ใน IP แพคเก็ต ที่เรียกว่าดาต้าแกรม การนำส่งข้อมูลของอุปกรณ์สื่อสารจึงใช้หลักการดูจากหมายเลขแอดเดรส IP ที่มีอยู่ในแพคเก็ต IP แล้วหาทางนำส่งไปให้ถูกต้อง การนำส่งดาต้าแกรมหรือ IP แพคเก็ตนี้ทำได้อย่างไร ทั้ง ๆ ที่เส้นทางของเครือข่ายสลับซับซ้อน

อุปกรณ์ที่ทำหน้าที่นำข้อมูลดาต้าแกรมนำส่งที่สำคัญคืออุปกรณ์หาเส้นทาง หรือที่เราเรียกว่าเราเตอร์ การทำงานของเราเตอร์จึงต้องมีโปรโตคอลพิเศษที่จัดการให้การเลือกเส้นทางเป็นไปอย่างมีประสิทธิภาพ การหาเส้นทางในกรณีนี้ใช้โปรโตคอลที่ช่วยสนับสนุนสำคัญสองประการคือ RIP-Routing Information Protocol และอีกโปรโตคอลหนึ่งที่กำลังนิยมคือ OSPF-Open Shortest Path First protocol

โมเดลของอินเทอร์เน็ตเป็นโมเดลเครือข่ายคอมพิวเตอร์ขนาดใหญ่ การทำงานจึงแบ่งแยกเป็นเครือข่ายย่อย ๆ หลาย ๆ เครือข่าย ที่แต่ละเครือข่ายทำงานกันเองอย่างอัตโนมัติ ดังตัวอย่างเช่น องค์กรองค์กรหนึ่งมีเครือข่ายก็จะทำงานเองอย่างอัตโนมัติ ผู้ดูแลและบริหารเครือข่ายแต่ละเครือข่ายก็จะตัดสินใจในการจัดการและโปรแกรมเราเตอร์ด้วยตัวเอง

คำถามต่อมาอยู่ที่ว่า เมื่อแต่ละเครือข่ายทำงานอิสระอย่างอัตโนมัติด้วยตนเองแล้ว จะนำมาร่วมกันได้อย่างไร

การกำหนดเส้นทาง IP แพคเก็ต

IP แพคเก็ตหรือดาต้าแกรม เดินทางตามเส้นทางผ่านอุปกรณ์กำหนดเส้นทางแต่ละตัวลำดับกันไป แต่ละตัวอาจเรียกเป็นโหนดเชื่อมโยงกันไปยังโหนดต่อ ๆ ไป โหนดที่อยู่ข้างเคียงเรียกว่าโหนดเพื่อนบ้าน

ในส่วนของแพคเก็ต IP มีส่วนหัวที่บ่งบอกข้อมูลแอดเดรส IP ต้นทางและแอดเดรส IP ปลายทาง และยังมีส่วนที่กำหนดเส้นทางโดยเรียกว่าตารางรายการกำหนดเส้นทาง (source route) เพื่อเป็นข้อมูลการกำหนดการเดินทางของ IP อย่างไรก็ดี ข้อมูลในส่วนหัวที่กำหนดเส้นทางก็มิได้หมายถึงว่าจะต้องกำหนดเส้นทางตามนั้นเสมอไป เพราะการกำหนดเส้นทางของ IP ในทางปฏิบัตินั้นอุปกรณ์เราเตอร์จะเป็นตัวกำหนดการทำงานให้เองตามความเหมาะสม

การกำหนดเส้นทาง ในระบบเครือข่ายคอมพิวเตอร์

การส่งข้อมูล IP ไปทางเกตเวย์หรือพอร์ตของเราเตอร์มีความสำคัญ เราเตอร์จะหาเส้นทางให้และตัดสินใจให้ว่าจะให้ดาต้าแกรมนั้นไปทางใด การปรับเปลี่ยนเส้นทาง หรือการเพิ่มเติมเส้นทาง การตัดเส้นทางเกิดขึ้นได้เสมอ เพราะเส้นทางสามารถโปรแกรมการเปลี่ยนแปลงได้ ทั้งที่เป็นการเปลี่ยนแปลงแบบชั่วคราวและถาวร บางครั้งการกำหนดเส้นทางก็แปรเปลี่ยนไปตามสถานะของสิ่งแวดล้อมต่าง ๆ

ตารางการกำหนดเส้นทาง

ในโฮสต์เหรือเราเตอร์ทุกตัวมีการรับและส่งข้อมูลระดับ IP ออกไปภายนอก การดำเนินการในลักษณะนี้กระทำโดยการดูจากตารางการกำหนดเส้นทาง (routing table)

การทำงานของโฮสต์หรือเราเตอร์มีวิธีการที่ง่าย ๆ กล่าวคือ เมื่อรับแพคเก็ต IP แล้วก็จะนำเอาข้อมูลแอดเดรส IP ปลายทางที่จะนำส่งมาเปรียบเทียบค่าตารางกำหนดเส้นทางเพื่อเลือกหาความเหมาะสมที่จะส่งออกทางด้านใด ตารางการกำหนดเส้นทางมิได้มีรูปแบบที่ตายตัว แต่ก็เป็นที่เก็บข้อมูลเส้นทาง ดังแสดงในรูปที่ 1

ข้อมูลที่เก็บไว้ในตารางการกำหนดเส้นทางประกอบด้วย แอดเดรสของเครือข่ายปลายทาง เครือข่ายย่อย (subnet) หรือโฮสต์ IP แอดเดรสของเราเตอร์ตัวที่ต่อถัดไป การอินเตอร์เฟสเครือข่ายที่ต่อร่วมเป็นอย่างไร ระยะทางที่จะไปถึงปลายทาง

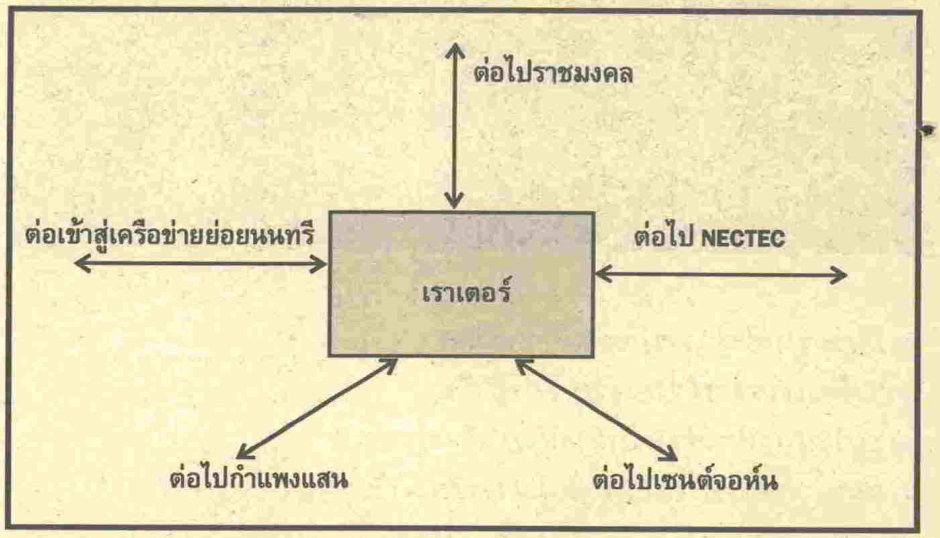
เพื่อให้การทำงานง่ายขึ้นจึงทำให้ตารางการกำหนดเส้นทางมีลักษณะสั้นและกะทัดรัด เพื่อว่าเมื่อต้องการเลือกเส้นทางจากตารางจะได้ดำเนินการได้รวดเร็ว จึงมักเก็บข้อมูลการกำหนดเส้นทางเป็นรหัสเครือข่ายหรือเครือข่ายย่อย

จากที่เรารู้มาบ้างแล้วว่าหมายเลข IP ใด ๆ เราแบ่งเป็นสองส่วนได้ โดยแยกเป็นรหัสเครือข่ายและรหัสโฮสต์ ดังนั้นการดำเนินการหาตำแหน่งโฮสต์จึงหาจากตำแหน่งเครือข่ายก่อนได้ แต่บางครั้งอาจมีความจำเป็นต้องเก็บรหัสโฮสต์ไว้ในตาราง

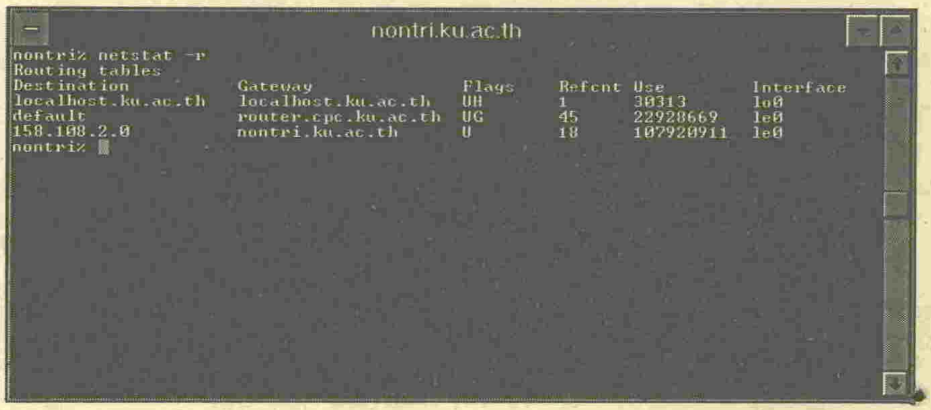
ในโฮสต์หรือเราเตอร์ทุกตัวที่เก็บตารางการกำหนดเส้นทางนี้ไว้ เราสามารถเรียกดูได้โดยใช้คำสั่ง netstat -r ลองดูตัวอย่างเมื่อเรียก netstat -r บนเครื่องยูนิกซ์ ดังแสดงในรูปที่ 2

ข้อมูลคอลัมน์แรกบอกถึงแอดเดรสของเครือข่ายปลายทาง จะเห็นว่าเรากำหนดในลักษณะเป็นเครือข่าย คอลัมน์ที่สองเป็นเส้นทางที่จะออกจากพอร์ต

บรรทัดแรกมีแอดเดรสปลายทางคือ localhost หรือ 127.0.0.1 และเกตเวย์ก็เป็น 127.0.0.1 กรณีนี้ 127 เป็นแอดเดรสในระบบที่มีไว้สำหรับให้กำหนดเป็นตัวเอง เพื่อที่จะทำลูปแบ็กเพื่อใช้ในการทดสอบ โดยการทดสอบจะส่งออกแล้วลองรับดูว่ารับข้อมูลได้ถูกต้องหรือไม่ คอลัมน์ Refcnt หมายถึงมีผู้ใช้กำลังใช้งานเชื่อมโยงทางเส้นทางนี้เท่าไร คอลัมน์ use บอกถึงจำนวนแพคเก็ตที่ส่งออกทางนี้ไปแล้วเป็นจำนวนกี่แพคเก็ต ส่วน Interfacelo0 บอกว่าเป็นอินเตอร์เฟสแบบลูปแบ็กอินเตอร์เฟส สำหรับ leo คือการ์ดอีเธอร์เน็ตที่อินเตอร์เฟสนั้นหมายถึงเส้นทางออกทางการ์ดนี้ ส่วน Flag เป็นการบอกว่าเกตเวย์ที่ต่อออกไปยังตัวเชื่อมโยงต่อไปมีสถานะเป็นอย่างไร เช่น เกตเวย์ router.cpc.ku.ac.th หมายถึงเราเตอร์ตัวต่อไป 158.108.2.1 นี้ มีสถานะเปิด หรือ on โดย Flags UG บอกถึงสถานะออน



รูปที่ 1 แสดงเราเตอร์ที่ต่อออกหลายเส้นทาง



รูปที่ 2

โปรโตคอลการกำหนดเส้นทาง

ตารางการกำหนดเส้นทางมาเก็บในโฮสต์หรือเราเตอร์ได้อย่างไร และมีวิธีการอย่างไรที่จะปรับปรุงข้อมูลส่วนนี้ให้ใช้ได้อยู่เสมอ หากมีทางเลือกในการกำหนดเส้นทางได้หลายทางเลือก ทางเลือกที่ดีที่สุดเป็นอย่างไร การกระทำในส่วนนี้จึงต้องมีโปรโตคอลในการกำหนดเส้นทาง โปรโตคอลในแบบที่ง่ายที่สุดเป็นดังนี้

ทำการศึกษาโครงข่ายของเครือข่ายคอมพิวเตอร์ที่คุณรับผิดชอบและดูเส้นทางการเชื่อมโยงต่าง ๆ

ใส่ตารางการกำหนดเส้นทางด้วยมือ

ปรับปรุงตารางการกำหนดเส้นทางด้วยมือ

ในกรณีนี้เหมาะสำหรับการทำงานของเครือข่ายขนาดเล็ก แต่หากเครือข่ายที่มีขนาดใหญ่และความซับซ้อนสูงขึ้น การปรับปรุงเปลี่ยนแปลงเกิดขึ้นได้เสมอ การป้อนตารางการกำหนดเส้นทางด้วยมือดูจะเป็นเรื่องเป็นไปไม่ได้ เพราะการทำงานจะมีเงื่อนไขเป็นตัวแปรอยู่ด้วย จึงยากที่จะให้บุคคลมาคอยติดตามและแก้ไขสถานการณ์ต่าง ๆ ให้ ในกรณีนี้จำเป็นต้องให้มีการทำงานแบบอัตโนมัติ โดยการแลกเปลี่ยนข่าวสารระหว่างเราเตอร์ด้วยกันเอง เกี่ยวกับข่าวคราวในเรื่องสถานะเครือข่าย และให้เราเตอร์แต่ละตัวปรับสถานะการทำงานที่ดีที่สุดของตนเอง

ผู้พัฒนาเราเตอร์ได้พยายามทำการค้นคว้าและวิจัยเกี่ยวกับวิธีการในการสร้างวิธีการกำหนดเส้นทางให้ดีที่สุด โดยมีการพัฒนาวิธีการเฉพาะขึ้นหลายอย่าง เช่น

- เมื่อเครือข่ายเปลี่ยนสถานะ ควรจะได้รับการรับรู้และปรับตัวให้เร็วที่สุด
- ควรมีวิธีการในการปรับเส้นทางโดยการคำนวณเลือกหาเส้นทางที่ดีที่สุด
- สามารถขยายเครือข่ายให้ได้อย่างต่อเนื่อง และเปลี่ยนรูปเครือข่าย
- ควรมีความละเอียดถี่ถ้วนในการใช้ทรัพยากรคอมพิวเตอร์
- ควรให้ความสำคัญและความละเอียดพิถีพิถันในการเลือกใช้ทรัพยากรการสื่อสารข้อมูล

การเลือกเส้นทางหรือการดำเนินการในเรื่องการกำหนดเส้นทางที่ดีจะทำให้การทำงานโดยรวมของเครือข่ายมีประสิทธิภาพ การออกแบบจึงต้องมองในภาพรวมเพื่อให้ได้สิ่งที่ต้องการ และต้องคำนึงถึงสิ่งต่าง ๆ ที่เกี่ยวข้องทั้งหมด

โปรโตคอล RIP

เพื่อให้การดำเนินการเลือกเส้นทางว่าทางใดดีกว่ากัน จึงกำหนดวิธีการจัดเส้นทางโดยการให้น้ำหนักของแต่ละเส้นทาง เช่น ถ้าเป็นเส้นทางเชื่อมระหว่างเราเตอร์กับเราเตอร์ที่เป็น LAN ให้น้ำหนักเป็น 1 แต่ถ้าเชื่อมด้วย WAN เช่น สาย 64 กิโลบิต ให้น้ำหนักเป็น 10 น้ำหนักที่มีทั้งหมดต้องไม่เกิน 15 หากมีค่าตั้งแต่ 16 ขึ้นไปถือว่าเส้นทางนั้นไปไม่ถึง

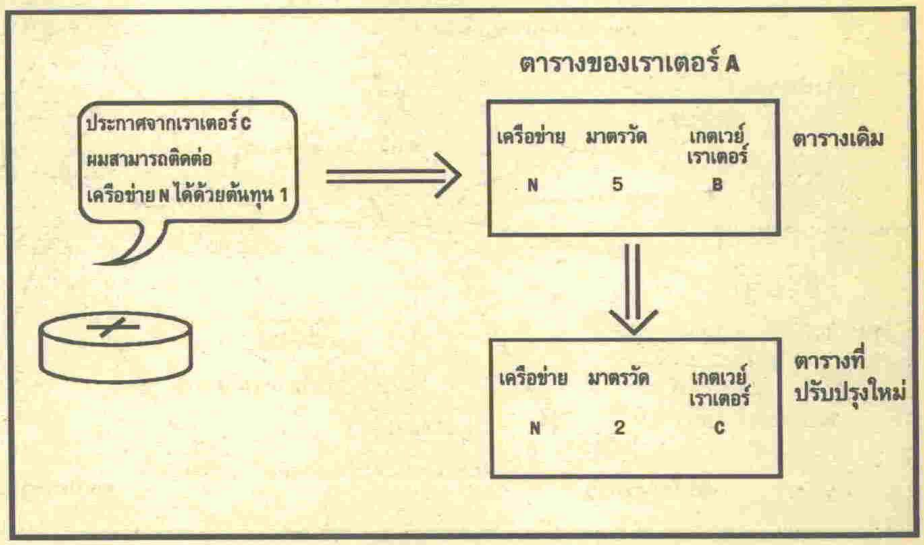
การกำหนดมาตรวัดนี้จึงมีอัลกอริธึมในการหาระยะทางโดยการคำนวณค่า ซึ่งเราเรียกว่า distance vector

อย่างไรก็ดี ในเราเตอร์ปัจจุบันมีการให้ความสำคัญในเรื่องนี้จึงมีการสร้างอัลกอริธึมเฉพาะมาช่วยคำนวณ โดยเรียกว่า link state routing algorithm การดูค่าการเชื่อมโยงมีการกำหนดรูปแบบและความเกี่ยวข้องหลาย ๆ อย่างร่วมด้วย เช่น ต้นทุนของค่าใช้จ่าย ความเชื่อถือ ความเร็ว และอื่น ๆ ประกอบกัน การดำเนินการเพื่อให้มีการเชื่อมโยงกันอย่างดีจึงมีการหาวิธีการเลือกเส้นทางที่เรียกว่า SPF หรือเส้นทางที่สั้นสุดก่อน (Shortest Path First)

โปรโตคอลที่ใช้ในการปรับปรุงวิธีการกำหนดเส้นทางและสร้างตาราง router ที่มีการใช้ตั้งแต่ยุคแรกคือ RIP RIP พัฒนาโดยบริษัทซีล็อก และมีการพัฒนาโปรแกรมการกำหนดเส้นทางนี้ไว้ในโปรแกรมสื่อสารในยูนิกซ์ RIP จึงขยายตัวและมีการใช้กันอย่างแพร่หลาย

การคำนวณหาเส้นทางและปรับปรุงตารางใช้วิธีการหาค่า distance vector มีการกำหนดเป็นค่า โดยตัวเลขสูงสุดคือ 15 ตัวเลขยิ่งต่ำแสดงว่าเส้นทางนั้นเป็นเส้นทางที่ดี มีต้นทุนต่ำ โดยปกติการเชื่อมโยงระหว่างเราเตอร์ตัวที่อยู่ติดกันมีต้นทุน 1

เพื่อให้ได้ข้อมูลที่ดีและเปลี่ยนแปลงอยู่เสมอ จึงกำหนดให้เราเตอร์ส่งข้อมูลข่าวสารกระจายออกไปยังเราเตอร์ตัวที่อยู่ข้างเคียงเพื่อการปรับปรุงข้อมูล การส่งนี้เป็นการส่งแบบกระจาย ทำการส่งกระจายออกไปทุก ๆ สามสิบวินาที และจะมีการนำข้อมูลนี้มาทำการปรับปรุงตารางภายใน



รูปที่ 3 เมื่อปรับปรุงตารางใหม่โดยใช้ข้อมูล RIP

เช่น เราเตอร์ A มีตารางของตัวเองโดยมีข้อมูลบอกว่าหากต้องการติดต่อกันเครือข่าย N จะส่งผ่านออกไปหาเราเตอร์ B โดยมีเวกเตอร์ของต้นทุน 5 ต่อมามีการกระจายข่าวจากเราเตอร์ C ว่า ตัวเองสามารถติดต่อกับเน็ตเวอร์ก N ได้โดยใช้ต้นทุนเพียง 1 เราเตอร์ A จะต้องมาทบทวนดูว่าการไปยังเครือข่าย N นั้นอาจผ่านเราเตอร์ C ไปได้ โดยรวมต้นทุนจากตัวเองไป C มีค่าเป็น 1 แล้วรวมต่อไป N มีค่าเป็น 2 ดังนั้นเราเตอร์ A จะพบว่าเส้นทางผ่าน C จะเป็นเส้นทางที่ดีกว่า จึงทำการเปลี่ยนแปลงตารางเพื่อให้การดำเนินงานไปยัง N ดีกว่าเดิม ดังแสดงในรูปที่ 3

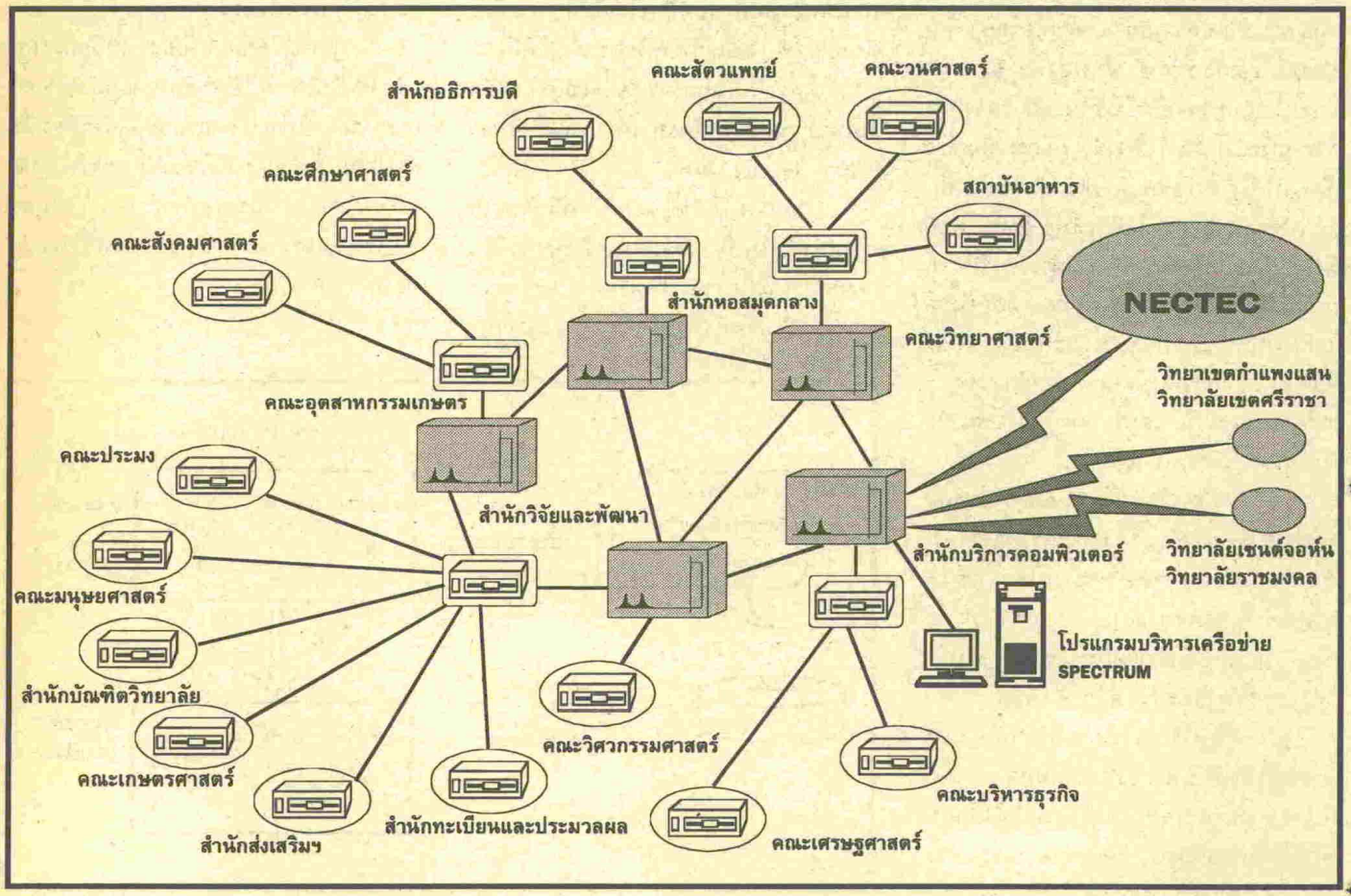
การกำหนดเส้นทางโดย RIP

คราวนี้ลองพิจารณาดูว่าถ้าหากเครือข่ายมีเราเตอร์หลาย ๆ ตัวและเชื่อมโยงกันด้วยลิงค์หลาย ๆ ลิงค์ เช่น เครือข่าย ku.ac.th มีการวางเราเตอร์ทั้งหมดเป็นเราเตอร์หลัก CISCO AGST 6 ตัว และเราเตอร์รองที่เป็น CISCO SIO อีกกว่า 10 ตัว จำนวนเราเตอร์ต่าง ๆ เหล่านี้เชื่อมโยงกันเป็นเครือข่ายที่มีเส้นทางเดินได้หลายทาง ดังตัวอย่างในรูปที่ 4 อย่างไรก็ดี ในขณะนี้กำลังดำเนินการระยะที่ 4 ซึ่งก็มีการเพิ่มขยายจำนวนเราเตอร์และปรับเปลี่ยนอีกหลายอย่าง

ถ้าสมมติว่าขณะนี้เราอยู่ที่โหนด i และต้องการระยะทางไปยังโหนดต่าง ๆ ในเครือข่าย จะกระทำอย่างไร ลองดูว่า RIP เข้ามาช่วยอย่างไร ทั้งนี้เพราะตัวที่อยู่ข้างเคียงจะส่งข้อมูลให้ระหว่างกันเพื่อปรับปรุงข้อมูล และมีการปรับเปลี่ยนสถานะกันไปเรื่อย ๆ เพราะทุกตัวรับข้อมูลและปรับเปลี่ยนกันด้วย ดังนั้นสถานะการปรับเปลี่ยนข้อมูลจึงกระทำไปเรื่อย ๆ

ลองดูขั้นตอนของโหนด i ที่จะปรับปรุงข้อมูลของตนเอง

1. เริ่มต้น เรารู้ระยะทางไปยังตัวที่อยู่ข้างเคียงเราทั้งหมด โดยกำหนดให้ตัวที่อยู่ข้างเคียงคือ k ดังนั้นระยะทางไปถึง k มีค่าเป็น d (i, k)



รูปที่ 4 เครือข่าย ku.ac.th ขณะเฟส 3

2. ตัวแปร D (m, n) หมายถึงระยะการรวมระหว่างโหนด m กับโหนด n ในขั้นต้นเราใส่ D (i, k)=d (i, k) ไปก่อนได้เลย โดยระยะทางไปยังโหนดอื่นในเครือข่ายถือว่ายังไปไม่ถึง คือใส่ค่าสูงสุดเป็น 16.7 ไว้

3. คราวนี้ถึงเวลาที่จะหาส่วนที่เหลือ เราจะได้รับข้อมูลจากตัวที่อยู่ข้าง ๆ เมื่อตัวที่อยู่ข้างคือ k ส่งข้อมูล D (k, j) มาให้ หมายถึงค่าจากโหนด k ไปยังโหนด j ดังนั้นเราสามารถนำมาคำนวณใหม่เป็นระยะทางจาก i ไป j ได้ดังนี้

$$D(i, j) = d(i, k) + D(k, j)$$

คือระยะทางรวมจาก i ไป j

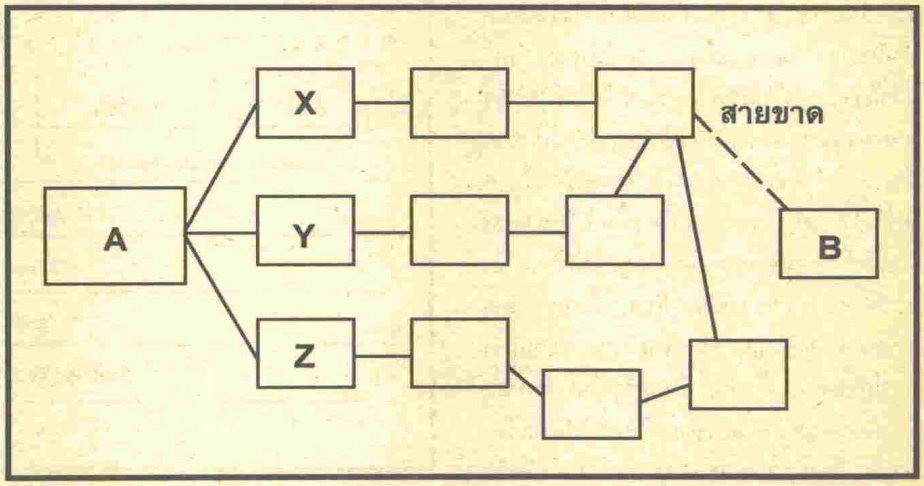
ให้ตรวจสอบดูว่า ถ้าส่วนนี้มีค่าน้อยกว่าที่มีอยู่ในตาราง ก็จะนำมาปรับปรุงคือนำมาเปลี่ยนแปลงไปจากเดิม หรือหากยังไม่มีก็เพิ่มเติมเข้าไป

การปรับปรุงข้อมูลนี้จะปรับปรุงกันทุกฝั่งจนได้ผลลัพธ์เป็นตารางการกำหนดเส้นทางที่มีประสิทธิภาพตารางหนึ่งในเราเตอร์

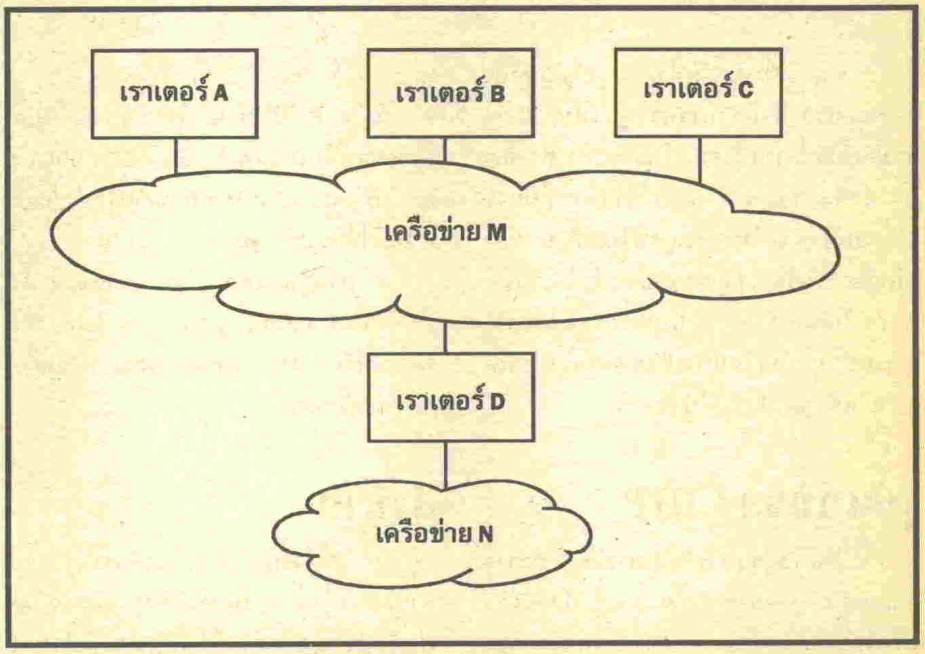
หากเมื่อมีเราเตอร์ปลายทางที่ไปไม่ถึงอาจต้องเสียเวลานานกว่าระบบอัตโนมัติจึงจะทำงานได้ครบทุกเราเตอร์ การดำเนินการต้องมีการปรับปรุงและแจ้งข่าวให้เร็วที่สุด มิเช่นนั้นการปรับปรุงระหว่างกันจะยังคงใช้ค่าต่ำไปปรับปรุง จึงเป็นผลเสีย

ลองดูจากตัวอย่างในรูปที่ 5 ซึ่งเราเตอร์ A ต้องการติดต่อกับ B เดิมเราเตอร์ A จะส่งข้อมูลผ่าน X แต่เมื่อไม่ถึง X ก็จะลองเปลี่ยนมาทาง Y ซึ่ง Y ยังไม่รู้ปัญหาว่าไปไม่ได้ และต่อมา Y ก็จะรู้ปัญหาว่าไปไม่ได้ จึงเปลี่ยนมาทาง Z

ระหว่างเวลาที่ A รู้ว่าตัวเองติดต่อ B ไม่ได้ก็ควรจะจบสิ้นและแจ้งข่าว กรณีนี้จะเห็นว่า RIP รู้ระยะทางไปยังโหนดข้างเคียง แต่ RIP ก็ไม่สามารถแจ้งข่าวร้ายในเครือข่ายได้รวดเร็ว ด้วยเหตุนี้การปรับปรุงข้อมูลระหว่างกันจึงต้องมีวิธีการที่ทำให้ได้ข้อมูลถูกต้อง รวดเร็ว และถึงกันทั้งหมด



รูปที่ 5 การปรับปรุงตารางเส้นทาง



รูปที่ 6 การแยกเราเตอร์กับเครือข่ายเพื่อให้การปรับปรุงตารางได้เร็ว

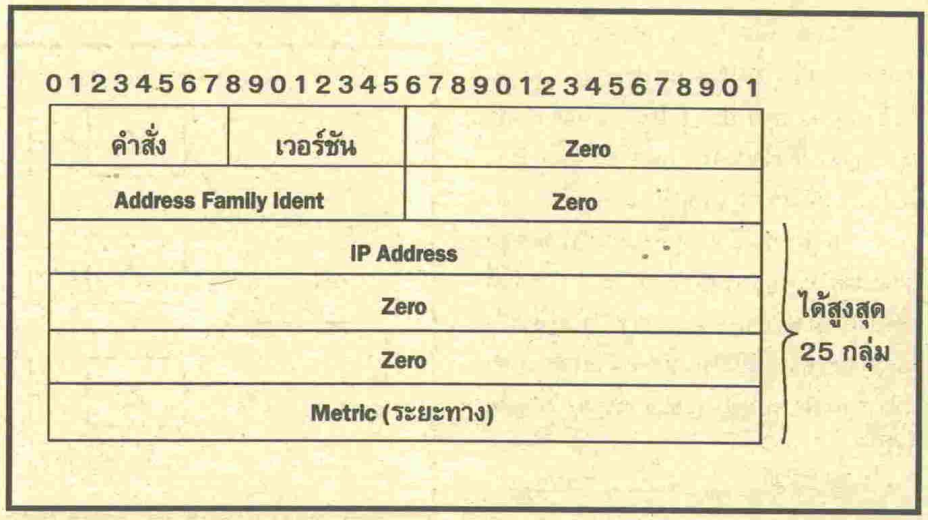
การตรวจสอบเราเตอร์ที่ไปไม่ถึงทำได้ด้วยการให้เราเตอร์แต่ละตัวคอยฟังข่าวสารจากเราเตอร์ตัวที่อยู่ข้าง ๆ โดยการคาดคะเนว่าจะต้องมีข่าวสารมาทุก ๆ 30 วินาที ถ้าหากเลยเวลา 180 วินาทีโดยไม่มีข่าวสารจากเราเตอร์ตัวข้าง ๆ มาถึง เราเตอร์จะมาร์กบอกว่าเราเตอร์ตัวที่อยู่ข้างนั้นไปไม่ถึงแล้ว

หลังจากนั้นเราเตอร์ก็จะส่งข้อความไปบอกตัวที่อยู่ถัดไปเสมือนการเตือนให้รู้ว่าเราเตอร์ตัวใดติดต่อไปไม่ได้ เพื่อให้รับรู้ข่าวสารนี้

การลดปัญหาที่เกิดขึ้นจากข่าวสารที่ส่งระหว่างกันเป็นเรื่องที่สำคัญ ลองดูตัวอย่างจากรูปที่ 6

จากรูปที่ 6 พบว่า เมื่อสายเชื่อมโยงระหว่างเราเตอร์ D กับเครือข่าย N ขาดจากกัน นั่นหมายความว่า เราเตอร์ D ไม่สามารถส่งข่าวสารข้อมูลไปยังเครือข่าย N ได้ แต่เราเตอร์ A ยังกระจายข่าวว่าตัวเองมีเส้นทางส่งไปเครือข่าย N ได้ ซึ่งเสมือนหลอกให้เราเตอร์ตัวอื่น ๆ เข้าใจผิด แม้เราเตอร์ D ที่ต่อบนเครือข่าย M ด้วยก็เข้าใจผิดจึงหาทางส่งข้อมูลไปทาง A ซึ่งจะสร้างปัญหา เราจึงแก้ปัญหาด้วยการมองเครือข่าย M เป็นแนวราบอันหนึ่ง หมายความว่าข้อมูลของ A ที่บอกมาบนเครือข่าย M นี้จะต้องให้ข้อมูลว่าตัวเองถ้าจะติดต่อไปยังเครือข่าย N ต้องผ่านเครือข่าย M และ D ซึ่งต่อกับเครือข่าย M บอกแล้วว่าไปยังเครือข่าย N ไม่ได้ ดังนั้นให้ทุกตัวบนเครือข่าย M รับรู้ว่าข้อมูลของ A นั้นใช้ไม่ได้ A จะปรับตัวเองต่อไป

เพื่อให้การปรับปรุงข้อมูลได้ถูกต้องและรวดเร็ว จึงมีวิธีการกระตุ้นให้ปรับปรุงข้อมูลกรณีมีการเปลี่ยนแปลงระยะทางที่สำคัญ การส่งข่าวสารจะส่งเฉพาะรายการที่มีการเปลี่ยนแปลง จะไม่ส่งข้อมูลที่ไม่เปลี่ยนแปลง เพื่อว่าการปรับปรุงโดยรวมจะทำได้เร็วขึ้น นอกจากนั้นยังมีกฎเกณฑ์บางอย่างเฉพาะเพื่อให้การทำงานในเรื่องการปรับปรุงข้อมูลไม่สับสนหรือทำสลับไปมาไม่รู้จบ



รูปที่ 7 รูปแบบของ RIP

รูปแบบของ RIP

ข้อมูลข่าวสารได้รับการส่งระหว่างเราเตอร์ในเครือข่ายตลอดเวลาอยู่แล้ว มีการทริกให้ปรับปรุงข้อมูล บางทีมีการส่งข้อความเพื่อขอให้ส่งข้อมูลกลับมายังเราเตอร์ โดยทั่วไปการขอให้ส่งข้อมูลมาจะกระทำระหว่างการเริ่มต้นระบบ หรือเมื่อทำตามฟังก์ชันเพื่อขอดูประสิทธิภาพเครือข่าย รูปแบบของ RIP สำหรับการขอและการตอบสนอง แสดงดังรูปที่ 7

กรณีที่คำสั่งเป็น 1 หมายถึงการขอ (request) และถ้าเป็น 2 หมายถึงการตอบสนอง (response) สำหรับเวอร์ชันเป็น 1 ส่วนของ Address Family Ident คือส่วนที่บอกว่าใช้โปรโตคอลการรับส่งในเครือข่ายเป็นโปรโตคอลอะไร ในที่นี้ใช้ IP สำหรับส่วนที่เหลือคือหมายเลข IP และ Metric คือการบอกรายละเอียดในรูประยะทางกับแอดเดรส โดยจะมีส่วนนี้ได้ถึง 25 กลุ่ม

สำหรับในฟอร์แมตการ request นั้นจะเป็นคำถาม ถ้าแอดเดรส 0 และ matric เป็น 16 หมายถึงการขอให้ปรับปรุงตารางการเลือกเส้นทางทั้งหมด

ส่งท้าย

โปรโตคอล RIP ใช้วิธีการปรับปรุงตารางการกำหนดเส้นทางให้ทันสมัย แต่โดยปกติแล้วในเครือข่ายที่ไม่ซับซ้อนก็เกือบจะไม่ต้องมีการปรับปรุงตารางอะไรเลย ดังนั้นการทำงานของอุปกรณ์กำหนดเส้นทางจึงทำงานตามตารางที่กำหนด เพราะตารางเกือบไม่มีการเปลี่ยนแปลง การทำงานของอุปกรณ์กำหนดเส้นทางจึงมีลักษณะเป็นแบบคงที่หรือที่เรียกว่าสแตติกเราต์

จากประสบการณ์ทั่วไปพบว่า เมื่อเครือข่ายที่มีโครงสร้างการทำงานซับซ้อนขึ้นจำเป็นต้องมีการกำหนดเงื่อนไขพิเศษสำหรับการเราต์ เพื่อให้ทำงานแบไดนามิกส์ RIP ก็ช่วยได้ส่วนหนึ่ง เช่น เมื่อเครือข่ายด้านหนึ่งเสียลงก็เลือกเส้นทางใหม่ได้ แต่ยังไม่สามารถหาเส้นทางที่ดีที่สุดให้ได้ การพัฒนาวิธีการหาเส้นทางจึงต้องพัฒนาต่อไปให้ดีขึ้น ในฉบับต่อไปผู้เขียนจะนำเสนอ OSPF ซึ่งเป็นโปรโตคอลที่ใช้กับเครือข่ายนนทรีซึ่งมีข้อดีในการทำงานอยู่มาก